

---

# LABORATÓRIO 24

---

## ARQUIVOS BINÁRIOS

### EXERCÍCIOS DE REVISÃO

---

VOCÊ DEVE ACOMPANHAR PARA OBTER INFORMAÇÕES COMPLEMENTARES

1. Construa um programa que leia um número desconhecido de peixes de um arquivo texto, chamado “pescado.txt”, e exiba a quantidade total de quilos de peixe pescado. O arquivo é formatado como no exemplo abaixo: o nome do peixe, o peso em gramas e o comprimento em centímetros.

```
Dourado 250 20
Tilapia 300 25
Carpa 400 40
Sardinha 50 10
Atum 150 15
```

2. Use o registro peixe definido abaixo.

```
struct peixe
{
    char nome[20];
    unsigned peso;
    float comp;
};
```

- a. Construa um programa que leia do usuário os dados de um único peixe e salve-os em um arquivo binário. Cada execução do programa deve acrescentar um peixe ao arquivo.
  - b. Modifique o programa obtido no item anterior para ler e exibir os peixes cadastrados no arquivo binário antes de permitir que o usuário faça um novo cadastro.
3. Construa um programa que leia um número desconhecido de peixes de um **arquivo texto** (semelhante ao arquivo utilizado na primeira questão) e grave estas informações em um **arquivo binário** que possa ser lido pelo programa da segunda questão. Compare o tamanho dos arquivos texto e binário.

## EXERCÍCIOS DE FIXAÇÃO

---

VOCÊ DEVE FAZER OS EXERCÍCIOS PARA FIXAR O CONTEÚDO

1. Escreva um programa que leia um valor inteiro do usuário e armazene-o em um arquivo binário chamado "integer.bin". Ao entrar, o programa deve verificar se o arquivo "integer.bin" existe, e em caso positivo, deve ler e exibir o número na tela.

Exemplo 1:

```
Digite um inteiro: 43
Número armazenado no arquivo integer.bin.
```

Exemplo 2:

```
Arquivo integer.bin encontrado!
Ele contém o número 43.

Digite um inteiro: 52
Número armazenado no arquivo integer.bin.
```

2. Construa um programa que leia do usuário uma sequência de números inteiros, até que o número zero seja lido. Os números devem ser armazenados em um arquivo binário chamado "vet.dat". O primeiro número do arquivo deve ser a quantidade de números digitados. Na entrada do programa verifique se o arquivo existe e, caso positivo, exiba os valores do vetor.

Exemplo 1:

```
Digite números (zero para encerrar):
50 82 45 31 94 22 0
6 números foram armazenados em vet.dat.
```

Exemplo 2:

```
0 arquivo vet.dat contém:
50 82 45 31 94 22

Digite números (zero para encerrar):
83 40 29 0
3 números foram armazenados em vet.dat.
```

**Dica:** como não sabemos de antemão quantos números serão digitados, podemos gravar o número zero como primeiro número do arquivo, gravar cada número digitado pelo usuário e depois retornar ao início do arquivo, com a função `seekp()` para sobrescrever o zero com o valor correto.

O comando abaixo posiciona o ponteiro de escrita do arquivo (`put`) na posição 0 a partir do seu início (`ios_base::beg`).

```
fout.seekp(0, ios_base::beg);
```

3. O Exercício de Revisão 1 do Laboratório 6 (Tipos Inteiros) mostra como usar códigos de Escape para mudar a cor do texto exibido em um terminal. A sequência de caracteres "\033[38;5;000;48;5;154m" define a cor 000 para o texto e 154 para o fundo. Considerando que as cores são valores de 3 dígitos de 000 a 255, construa um programa para ler do usuário um texto com no máximo 80 caracteres, um código para a cor do texto e um código para a cor do fundo. Guarde essas informações em um arquivo binário. O usuário deve ter a opção de guardar uma nova frase ou exibir a frase já armazenada através de um menu, como mostrado no exemplo abaixo.

```
Mensagens Coloridas
-----
[A]rmazenar
[E]xibir
[S]air
-----
Opção: [A]

Sua mensagem: Feliz Aniversário!
Cor do texto: 130
Cor do fundo: 038

Texto colorido foi armazenado.
```

4. No exercício anterior, se o usuário digitar apenas espaços para a mensagem, o resultado será uma faixa colorida com a cor de fundo. Podemos usar isso para criar uma imagem formada apenas por caracteres de espaço coloridos.

Construa um programa que peça a largura e altura da imagem e leia do usuário uma matriz de números. Cada número de 3 dígitos representa a cor de um "bloco" da imagem. Guarde em um arquivo binário os valores de altura, largura e de cada bloco da imagem. Assim como o programa anterior, construa um menu para controlar as opções de armazenamento e exibição da imagem.

```
Imagens Coloridas
-----
[A]rmazenar
[E]xibir
[S]air
-----
Opção: [A]

Largura: 5
Altura : 5

Defina 5 cores em cada linha
#1: 220 220 220 220 220
#2: 220 000 220 000 220
#3: 220 220 220 220 220
#4: 220 000 000 000 220
#5: 220 220 220 220 220

Imagem colorida foi armazenada.
```

## EXERCÍCIOS DE APRENDIZAGEM

---

VOCÊ DEVE ESCREVER PROGRAMAS PARA REALMENTE APRENDER

1. Crie um programa que grave os 100 primeiros números naturais ao mesmo tempo em um arquivo texto e em um arquivo binário. Compare os tamanhos dos arquivos e explique como os tipos escolhidos para guardar os números podem tornar o arquivo binário menor ou maior que o arquivo texto.
2. Crie um registro Soldado com os campos nome, eliminações, mortes, taxa de eliminações por morte (eliminações/mortes) e número de partidas jogadas. Na função principal leia um soldado de um arquivo binário. Se o arquivo não existir, você deve direcionar o usuário para a criação um novo soldado. Quando o soldado for lido (seja pelo arquivo ou pelo teclado), você deve dar as seguintes opções ao usuário:

```
[N]ovo soldado (sobrescreve o anterior)
[A]tualiza soldado (com os dados da última partida)
[E]xibe soldado (atual)
[S]air
```

**Novo:** deve pedir as informações de um soldado, armazená-las em memória durante a execução do programa ao final do programa gravá-las em um arquivo binário.

**Atualiza:** o usuário deve digitar as informações referentes a última partida do jogo (número de eliminações e mortes), e o programa deve atualizar as informações do soldado contabilizando as informações dessa partida.

**Exibe:** crie uma função que receba um Soldado e mostre as informações dele na tela. Chame a função passando o soldado atual sempre que essa opção for selecionada.

3. Crie uma união chamada Senha com os campos Alfanumérica e Numérica. Na função principal pergunte ao usuário se ele quer exibir a senha armazenada ou gravar uma nova senha no arquivo. Se o arquivo não tiver sido criado ainda, a opção de exibição não deve fazer nada, a não ser avisar o usuário que o arquivo não existe.

Use um arquivo binário para guardar a senha. Na gravação da senha, grave primeiro um número para representar o tipo da senha. O modo alfanumérico será correspondente ao número 1, e o modo simplesmente numérico será correspondente ao 2. Quando o usuário abrir o programa e selecionar a opção de exibição, o tipo da senha deve ser lido para decidir qual campo da união apresentar na tela (campo alfanumérico ou o campo numérico).